

# CIS 320

## Introduction to Software Engineering

### Spring 2018



**Professor:** Dr. Joe Hoffert  
joe.hoffert@indwes.edu

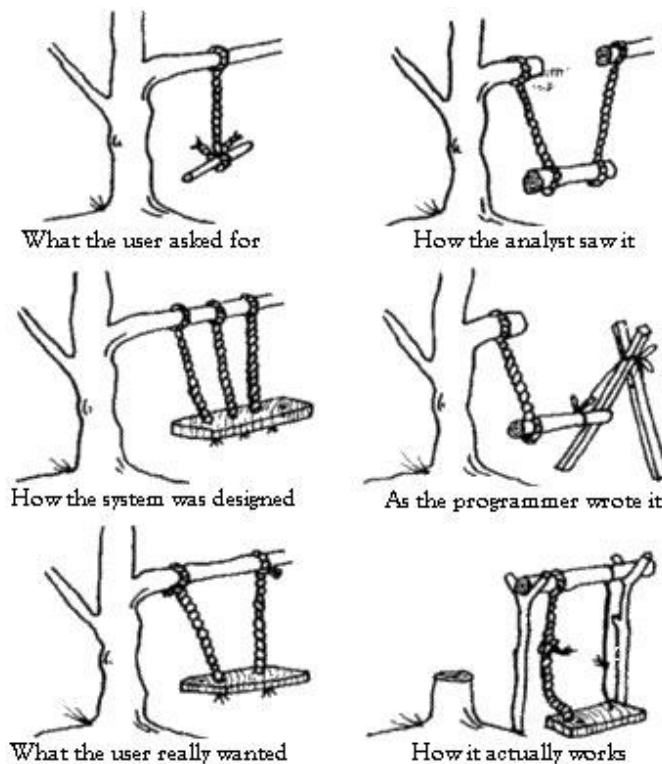
**Credits:** 3

**Class Times:** MWF Ott 182 1:25 - 2:20

**Office & Hours:** Ott 170-C

**Telephone:** 765-677-2292

---



**Some of the multiple perspectives and potential problems in developing software**

## Course Philosophy:

*Education is not about the filling of a bucket but the lighting of a fire!* --William Butler Yeats

*Who questions much, shall learn much, and retain much.* --Francis Bacon

*Good judgment comes from experience, and experience comes from bad judgment.* --Frederick P. Brooks

*Own your education.* --Joe Hoffert

*It is our duty and joy at **all times** and in **all places** to give You thanks and praise, Holy Father, heavenly King, Almighty and eternal God.* -- The Liturgy of the Church Of Nigeria (Anglican Communion)

[Sharing an Adventure](#)

[Educational Paradigm](#)

[Software Engineering Challenge Question](#)

---

## Prerequisites

- CIS-221 and CIS-222 (or one as co-requisite).
- 

## I. Course Description

This course provides an introduction to the process of developing large scale software systems. Included is the study of the procedures and tools of software engineering. Concepts introduced are software life-cycle models, quality factors, requirements analysis and specification, software design (functional design and object-oriented design), implementation and testing.

## II. Course Objectives

After taking this course, the students should have:

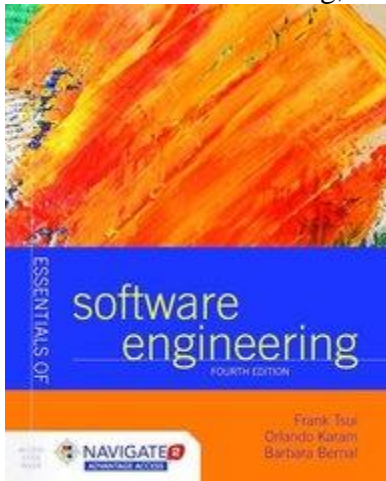
- Gained an understanding of software engineering principles;
- Explored such topics as systems life cycle models, requirements definitions and analysis, specifications, software testing techniques and verification and validation, software quality factors;
- Considered the goals and requirements of software project management;

- Completed an overview of current software methodologies and design practices; and
- Considered how the practice of software engineering and its associated implications in ethics and professionalism may be practiced by Christians intent on changing their world

### III. Required Text and Materials

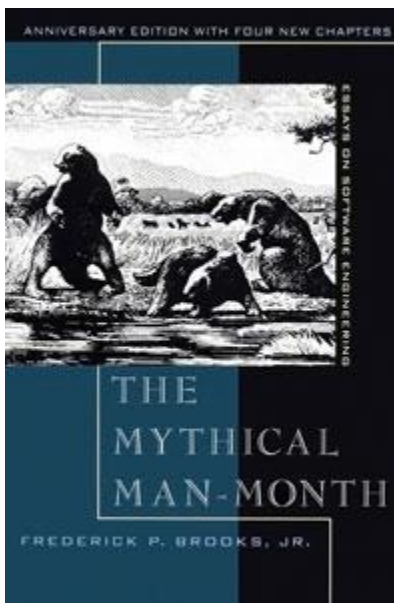
Required Text:

Essentials of Software Engineering (4<sup>th</sup> ed.): Frank Tsui, Orlando Karam, Barbara Bernal; Jones & Bartlett Learning, ISBN 9781284106008.



Recommended Text:

The Mythical Man-Month, Fred Brooks, Addison-Wesley, ISBN 0-201-83595-9.



### IV. Grade Distribution

<b>Weekly Hours &amp; Unplugging</b>	<b>65 pts</b>	<b>Grading Scale</b>
<b>Participation</b>	<b>115 pts</b>	<b>A: 94 – 100%</b>
<b>Presentations</b>	<b>120 pts</b>	<b>A-: 90 – 93%</b>
<b>Midterm Examination</b>	<b>100 pts</b>	<b>B+: 87 – 89%</b>
<b>Final Examination</b>	<b>150 pts</b>	<b>B: 83% – 86%</b>
<b>Project</b>	<b>450 pts</b>	<b>B-: 80 – 82%</b>
	<hr/>	<b>C+: 77 – 79%</b>
<b>Total</b>	<b>1000 pts</b>	<b>C: 73 – 76%</b>
		<b>C-: 70 – 72%</b>
		<b>D+: 67 – 69%</b>
		<b>D: 63 – 66%</b>
		<b><u>F: 0% – 62%</u></b>

## V. Description of Assignments

1. **Weekly Hours:** At the end of each week (*i.e.*, by Saturday 11:59 p.m.), students are to submit the number of hours they spent on this course outside of class time. These time submissions are graded only in that any submission with the information will receive the full 5 weekly points. For example, if one student reports that he or she spent zero (0) hours on the course outside of class and another reports that he or she spent 15 hours on the course outside of class, both will receive the 5 weekly points.

The expected average time a student should spend on any course is 2 hours out of class for every hour in class. For a 3 hour class that maps to 6 hours per week *on average* outside of class. **You can expect to spend more time than this in courses directly related to your major** (which this course likely is).

2. **Attendance:** The classroom times are meant to be interactive discussions eliciting comments, questions, and concerns. The expectation for the course is that students will not only attend but participate in the discussions **which will be especially true in this project-based group work class**. For chapters 1-3 and 5, the night before class students will submit three questions (in Brightspace) relevant to the reading assignment. This approach is meant to foster learning in a group environment. *Chapter quizzes* which will be given after covering each of these first chapters in class. The material covered on the quizzes will be taken directly from the classroom discussions, the assignments, and the textbook. The quizzes are meant for formative assessment where you evaluate what you think you have learned already (rather than summative assessment where you are evaluated on what you are expected to know). The quizzes are good study guides for exams.

**After completing Chs. 1-3 & 5, students will be reading chapters (*i.e.*, Chs. 4 & 6-11) and discerning how they relate to the project work they are doing. Rather than submitting questions about the chapter students will write a minimum of 250**

words total about two aspects from the chapter and how they relate or don't relate to their project work and why. Each aspect needs to include a heading or subheading (*e.g.*, subheading 4.2.1 relates to the Waterfall Model). These write-ups will be due once a week. These write-ups also relate to the final exam as outlined below.

3. **Presentations:** During the course of the semester each pair of students will make a presentation on any topic relevant to software engineering and each project group will make a presentation at a division colloquium or a Ft. Wayne Tech Meet-up. Any topic related to the textbook, exercises, or class discussions is valid for an individual presentation. Other topics will need the approval of the instructor. Each student pair is strongly urged to pick a topic that is of interest to them. These presentations provide the students with the opportunity to become familiar with research that is being done regarding software engineering as well as providing oral and written presentation practice. [The presentation schedule is available.](#)

**[100 pts]** Each pair of students should a topic related to software engineering that is of interest to them. The students should research the topic to find out more about it and present the findings to the class. In particular, the following questions should be addressed:

- What is the problem being addressed?
- How is that problem relevant to software engineering in general and to the students' project(s) in particular?
- Why is the problem important (or not)?
- Why is the problem/topic considered interesting to the presenter?
- What research is being done regarding the problem?

Each student pair presentation is expected to last approximately 10 minutes with 5 minutes afterwards for questions. For each presentation the students will need to incorporate and reference a peer-reviewed research paper. Two web sites available are the [ACM website](#) and the [IEEE Xplore website](#). *Other sources may be used with permission from the instructor.* If you are not able to access the full text of the papers contact me for assistance. [A rubric for the write-ups and presentations can be found here.](#) Using a presentation software package is appropriate (*e.g.*, Microsoft PowerPoint, OpenOffice Impress) but not required. Other forms of disseminating information (*e.g.*, videos, poems, dramas, video games) are also encouraged **and have the possibility of earning extra credit.**

Each student pair will also need to write a 3 full-page paper relevant to the selected topic. The paper can reiterate the presentation or provide more in depth coverage of the topic.

As part of the grade for each write-up/paper each pair of students will need to schedule and attend a session at the Inkwell or Inkspot to cover areas of writing

such as (but not limited to) citations, punctuation, grammar, transitions, introduction, and conclusion. 20% of the write-up grade will be deducted if proof of an Inkwell session is not included. (More information about Inkwell should be available from the “Student Resources and Information” link in BrightSpace.)

**[20 pts] Additionally,** each *project* group will make a 30 minute presentation at a division colloquium or a Ft. Wayne Tech Meetup during the semester. Obviously, the project doesn’t need to be completed. However, with the development approach taken, after every 2 week development cycle a viable product is expected. This product and the accompanying user stories should be used for the presentation.

4. **Projects:** There are two types of projects in which each student will be involved. One is an individual project that will be formative in nature (*i.e.*, develop the project with what you know and are getting from the course). The student can pick one of [two projects](#) to work on for the first few weeks of the course. At the end of each of the first few weeks, the student will submit a [Formative Weekly Checklist](#). For the **first Saturday** in the course, you should turn in your project selection and your time estimation (*i.e.*, how many hours of work you think the project will take you to complete).

The other project is group based and will be more summative in nature (*i.e.*, develop the project based on what is in the textbook, what we’ve discussed in class, and the available rubric). The software engineering approach for this project will be based on agile development strategies (*cf.* chapter 5 in the textbook) with a [two-week development cycle](#). Student groups will select a project to work on and present at the end of the course. Groups can select from [several project opportunities](#) or create their own project. This project will provide opportunities for students to discuss, explore, and implement the topics addressed in the classroom via group work culminating in a presentation at the end of the course. The groups will present status updates during class times using the [bi-weekly development cycle rubric](#). [A rubric for software coding is available](#). **NOTICE:** Projects will be group work and graded by group. However, group members will evaluate each other. [The presentation schedule \(with groupings\) is available](#).

5. **Midterm Examination:** The midterm examination is scheduled for **Monday, Feb. 19** in Ott 182 during the normal time starting at 1:25. The midterm examination will be based on material from the classroom discussions, textbook, and quizzes.
6. **Final Examination (During Exam Week):** The final examination schedule is **Tuesday, Apr. 24, 2018** between **2:00—3:50**. The final examination will be open book and cover the chapters since the mid-term exam. The final examination will be based on the write-ups from the last chapters (*i.e.*, Chs. 4, 6-11) in that each student will need to address two areas/headings/subheadings for each chapter.

## VI. Preliminary Topical Outline for the Course

<u>Topic</u>	<u>Reading</u>	<u>Week</u>	<u>Finish 1<sup>st</sup> 1/2 By</u>	<u>Finish Reading By</u>	<u>Assignment Due</u>
<i>Intro to Course; Writing a Program</i>	<i>Chapter 1</i>	<i>1</i>	_____	<i>Jan. 9</i>	Formative Weekly Checklist [10 pts] (due Jan. 13, 11:59 p.m.)
[Martin Luther King, Jr. Day Jan. 15 – no class] <i>Building a System (Faculty candidate Dr. Andre Fonteles in charge Friday, Jan. 19)</i>	<i>Chapter 2</i>	<i>2</i>	_____	<i>Jan. 16</i>	Formative Weekly Checklist [10 pts] (due Jan. 20, 11:59 p.m.)
<i>Engineering of Software</i>	<i>Chapter 3</i>	<i>3</i>	_____	<i>Jan. 23</i>	Formative Weekly Checklist [10 pts] (due Jan. 27, 11:59 p.m.)
<b>New and Emerging Process Methodologies</b>	<b>Chapter 5</b> <b>(Note: order switch)</b>	<b>4</b>	<b>Jan. 27</b>	<b>Jan. 30</b>	
<b>Software Process Models</b>	<b>Chapter 4</b>	<b>5</b>	_____	<b>Feb. 6</b>	Formative Weekly Checklist [10 pts] Project: Application Approval [10 pts] (due Feb. 3, 11:59 p.m.)
<b>Requirements Engineering</b>	<b>Chapter 6</b>	<b>6</b>	_____	<b>Feb. 13</b>	Project: 1 <sup>st</sup> Bi-weekly Status [100 pts] (due Feb. 13 in class)
<b>Midterm Exam</b> <b>[Feb. 19]</b> <b>Presentations</b> <b>[Feb. 21 &amp; 23]</b>	_____	<b>7</b>	_____	_____	

No reading due this week; No classes this week due to Ontario Systems field trip		8			
<b>Design: Architecture and Methodology</b>	<b>Chapter 7</b>	<b>9</b>	-----	<b>March 13</b>	
<b>Design Characteristics and Metrics</b>	<b>Chapter 8</b>	<b>10</b>	-----	<b>March 20</b>	Project: 2 <sup>nd</sup> Bi-weekly Status [100 pts] (due March 23 in class)
<b>Implementation</b>	<b>Chapter 9</b>	<b>11</b>	-----	<b>March 27</b>	
<b>Testing and Quality Assurance</b>	<b>Chapter 10</b>	<b>12</b>	-----	<b>April 3</b>	Project: 3 <sup>rd</sup> Bi-weekly Status [100 pts] (due April 6 in class)
<b>Configuration Management, Integration, and Builds</b>	<b>Chapter 11</b>	<b>13</b>	-----	<b>April 10</b>	
<b>Project Presentations (April 20 &amp; 23)</b>	-----	13	-----	-----	Final Project Presentations [100 pts] (due April 20 & 23 in class)

**VII. Late work: Work turned in after the due date & time will receive a mark of 0 (unless there are extenuating circumstances, e.g., death of a nuclear family member).** This policy is meant to help the students in regards to accountability.

**VIII. Academic Honesty:**

**Do not share or copy work. You are encouraged to discuss ideas, approaches, comments, etc. but your work must be your own. You should type in all the code yourself. Do not copy and paste from another source. You may copy and share code in your project groups.**